

Amendments to the Claims:

This listing of claims will replace all prior versions and listing of claims in the parent PCT patent application for purposes of the national phase application in the United States. In this preliminary amendment, claims 4, 10, and 16 have been amended. Status identifiers provide an indication of the status of the claims relative to the claims listed in the International Preliminary Examination Report (IPER) of the corresponding parent PCT patent application.

Claim 1. (original) A device comprising:

- a memory unit including executable software;
- a plurality of class files stored in the memory unit; and,
- a computing unit connected to the memory unit, the computing unit being able to execute a Java Virtual Machine, the computing unit executing the executable software for generating a number of cod files from the plurality of class files by combining elements from the plurality of class files without duplication of entries for reducing storage space, wherein the number of cod files is less than the number of class files and a given cod file includes:
 - a constant pool created by combining constant pool entries from two or more of the class files without duplication of entries;
 - a byte codes and information structure created by combining byte codes and information structure entries from the two or more of the class files without duplication of entries; and,
 - a fixup table for providing information to the Java Virtual Machine for resolving at least one entry in the given cod file at link time.

Claim 2. (original) The device of claim 1, wherein for the given cod file, the fixup table includes a symbolic reference for cross-referencing a method not contained in the cod file.

Claim 3. (original) The device of claim 1, wherein the given cod file further includes a sibling list for listing other related cod files to define a sibling group and the fixup table of the given cod file further includes indices to the other related cod files specified in the sibling group.

Claim 4. (currently amended) The device of ~~any preceding~~ claim 1, wherein for the given cod file, the byte codes and information structure includes a hard offset for cross-referencing a method included in the given cod file that was previously symbolically referenced.

Claim 5. (original) The device of claim 3, wherein for the given cod file, the byte codes and information structure includes a hard offset for cross-referencing a method included in the sibling group that was previously symbolically referenced.

Claim 6. (original) The device of claim 3, wherein for the given cod file, the fixup table includes a symbolic reference for cross-referencing a method not contained in the sibling group.

Claim 7. (original) A method for generating a number of cod files from a plurality of class files by combining elements from the plurality of class files such that the number of cod files is less than the number of class files without duplication of entries for reducing storage space, wherein for a given cod file, the method comprises:

- identifying class files with common entries in at least one of the constant pool and the byte codes and information structure;

- generating a constant pool for the given cod file by combining constant pool entries from the class files with common entries without duplication;

- generating the byte codes and information structure for the given cod file by combining byte codes and information structure entries from the class files with common entries without duplication; and,

generating a fixup table for providing information to a Java Virtual Machine for resolving at least one entry in the given cod file at link time.

Claim 8. (original) The method of claim 7, wherein the method further includes, for the given cod file, providing a symbolic reference in the fixup table for cross-referencing a method not contained in the cod file.

Claim 9. (original) The method of claim 7, wherein the method further includes:
generating a sibling list for listing other related cod files to define a sibling group; and,
providing indices to the other related cod files specified in the sibling group in the fixup table.

Claim 10. (currently amended) The method of ~~any preceding claim 7~~, wherein the method further includes, for the given cod file, providing a hard offset in the byte codes and information structure for cross-referencing a method included in the given cod file that was previously symbolically referenced.

Claim 11. (original) The method of claim 9, wherein the method further includes, for the given cod file, providing a hard offset in the byte codes and information structure for cross-referencing a method included in the sibling group that was previously symbolically referenced.

Claim 12. (original) The method of claim 9, wherein the method further includes, for the given cod file, providing a symbolic reference for the fixup table for cross-referencing a method not contained in the sibling group.

Claim 13. (original) An article storing executable software that when executed by a computing unit generates a number of cod files from a plurality of class files by combining elements from the plurality of class files without duplication of entries

for reducing storage space, wherein the executable software comprises code for generating a given cod file to include:

- a constant pool created by combining constant pool entries from two or more of the class files without duplication of entries;

- a byte codes and information structure created by combining byte codes and information structure entries from the two or more of the class files without duplication of entries; and,

- a fixup table for providing information to a Java Virtual Machine for resolving at least one component of the given cod file at link time.

Claim 14. (original) The article of claim 13, wherein for the given cod file, the fixup table includes a symbolic reference for cross-referencing a method not contained in the cod file.

Claim 15. (original) The article of claim 13, wherein the given cod file further includes a sibling list for listing other related cod files to define a sibling group and the fixup table of the given cod file further includes indices to the other related cod files specified in the sibling group.

Claim 16. (currently amended) The article of ~~any preceding claim~~ 13, wherein for the given cod file, the executable software comprises code for generating a hard offset in the byte codes and information structure for cross-referencing a method included in the given cod file that was previously symbolically referenced.

Claim 17. (original) The article of claim 15, wherein for the given cod file, the executable software comprises code for generating a hard offset in the byte codes and information structure for cross-referencing a method included in the sibling group that was previously symbolically referenced.

Claim 18. (original) The article of claim 15, wherein for the given cod file, the executable software comprises code for generating a symbolic reference in the fixup table for cross-referencing a method not contained in the sibling group.